CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

INSTITUTE : UIE

DEPARTMENT : CSE

Bachelor of Engineering (Computer Science & Engineering)

PROJECT BASED LEARNING IN JAVA

(20CST-319/20ITT-319)

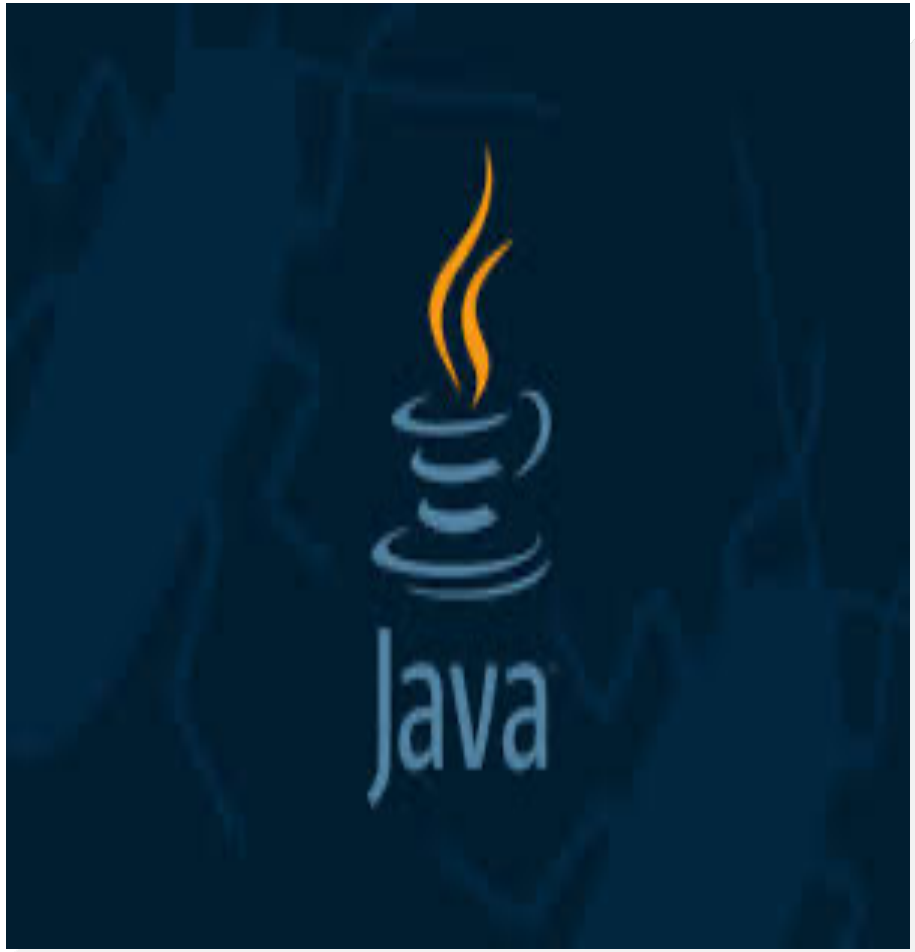
TOPIC OF PRESENTATION:

Use of public, private and protected.

Lecture Objectives

In this lecture, we will discuss:

•Use of public, private and protected.



To understand the concept of Access Specifier, you must have the knowledge of packages in java.

A package as the name suggests is a pack(group) of classes, interfaces and other packages. In java we use packages to organize our classes and interfaces.

We have two **types of packages in Java**: built-in packages and the packages we can create (also known as user defined package).

- 1) User defined package: The package we create is called user-defined package.
- 2) Built-in package: The already defined package like java.io.*, java.lang.* etc are known as built-in packages.

Sub packages in Java

- A package inside another package is known as sub package. For example If we create a package inside letmecalculate package then that will be called sub package.
- Lets say we have created another package inside letmecalculate and the sub package name is multiply. So if we create a class in this subpackage it should have this package declaration in the beginning:

package letmecalculate.multiply;

Points to remember:

1. Sometimes class name conflict may occur. For example: Lets say we have two packages **abcpackage** and **xyzpackage** and both the packages have a class with the same name, let it be JavaExample.java. Now suppose a class import both these packages like this:

```
import abcpackage.*;

import xyzpackage.*;
```

This will throw compilation error. To avoid such errors you need to use the fully qualified name method that I have shown above. For example

```
abcpackage.JavaExample obj = new abcpackage.JavaExample();

xyzpackage.JavaExample obj2 = new xyzpackage.JavaExample();
```

This way you can avoid the import package statements and avoid that name conflict error.

2. If we create a class inside a package while importing another package then the package declaration should be the first statement, followed by package import. For example:

```
package abcpackage;

import xyzpackage.*;
```

3. A class can have only one package declaration but it can have more than one package import statements. For example:

```
package abcpackage; //This should be one

import xyzpackage;

import anotherpackage;

import anything;
```

4. The wild card import like package.* should be used carefully when working with subpackages. For example: Lets say: we have a package **abc** and inside that package we have another package **foo**, now **foo** is a subpackage.

classes inside abc are: Example1, Example 2, Example 3

classes inside foo are: Demo1, Demo2

So if I import the package **abc** using wildcard like this:

import abc.*;

Then it will only import classes Example1, Example2 and Example3 but it will not import the classes of sub package.

To import the classes of subpackage you need to import like this:

import abc.foo.*;

This will import Demo1 and Demo2 but it will not import the Example1, Example2 and Example3.

So to import all the classes present in package and subpackage, we need to use two import statements like this:

import abc.*;

import abc.foo.*;

Java Access Modifiers – Public, Private, Protected & Default

An access modifier restricts the access of a class, constructor, data member and method in another class. In java we have four access modifiers:

1. default
2. private
3. protected
4. public

1. Default access modifier

When we do not mention any access modifier, it is called default access modifier. The scope of this modifier is limited to the package only.

This means that if we have a class with the default access modifier in a package, only those classes that are in this package can access this class. No other class outside this package can access this class.

Similarly, if we have a default method or data member in a class, it would not be visible in the class of another package.

2. Private access modifier

The scope of private modifier is limited to the class only.

Private Data members and methods are only accessible within the class

Class and **Interface** cannot be declared as private

If a class has **private constructor** then you cannot create the object of that class from outside of the class.

3. Protected Access Modifier

Protected data member and method are only accessible by the classes of the same package and the subclasses present in any package. You can also say that the protected access modifier is similar to default access modifier with one exception that it has visibility in sub classes.

Classes cannot be declared protected. This access modifier is generally used in a parent child relationship.

4. Public access modifier

The members, methods and classes that are declared public can be accessed from anywhere. This modifier doesn’t put any restriction on the access.

The scope of access modifiers in tabular form

	Class	Package	Subclass (same package)	Subclass (diff package)	Outside Class
public	Yes	Yes	Yes	Yes	Yes
protected	Yes	Yes	Yes	Yes	No
default	Yes	Yes	Yes	No	No
private	Yes	No	No	No	No

QUIZ:

1. Which of these access specifiers must be used for main() method?
   a) private
   b) public
   c) protected
   d) none of the mentioned

2. Which of the following statements are incorrect?
   a) public members of class can be accessed by any code in the program
   b) private members of class can only be accessed by other members of the class
   c) private members of class can be inherited by a subclass, and become protected members in subclass
   d) protected members of a class can be inherited by a subclass, and become private members of the subclass

Summary:

In this session, you were able to :

- Learn about Use of public, private and protected.

References:

Books:

1. Balaguruswamy, *Java.*
2. A Primer, E.Balaguruswamy, *Programming with Java,* Tata McGraw Hill Companies
3. John P. Flynt Thomson, *Java Programming.*

Video Lectures :

https://www.youtube.com/watch?v=eEujVn-ZTLE

Reference Links:

https://www.geeksforgeeks.org/packages-in-java/

https://www.javatpoint.com/package

THANK YOU